Information Processing and Management 51 (2015) 340–358

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

Automatic summarization assessment through a combination of semantic and syntactic information for intelligent educational systems

Asad Abdi [a,*], Norisma Idris [a], Rasim M. Alguliev [b], Ramiz M. Aliguliyev [b]

[a] Department of Artificial Intelligence Faculty of Computer Science and Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia
[b] Institute of Information Technology, Azerbaijan National Academy of Sciences, 9, B. Vahabzade Street, AZ1141 Baku, Azerbaijan

ARTICLE INFO

Article history:
Received 22 March 2014
Received in revised form 22 January 2015
Accepted 3 February 2015

Keywords:
Automatic summary assessment
Content coverage
Natural language processing
Automatic grading
Intelligent tutoring systems

ABSTRACT

Summary writing is a process for creating a short version of a source text. It can be used as a measure of understanding. As grading students' summaries is a very time-consuming task, computer-assisted assessment can help teachers perform the grading more effectively. Several techniques, such as BLEU, ROUGE, N-gram co-occurrence, Latent Semantic Analysis (LSA), LSA_Ngram and LSA_ERB, have been proposed to support the automatic assessment of students' summaries. Since these techniques are more suitable for long texts, their performance is not satisfactory for the evaluation of short summaries. This paper proposes a specialized method that works well in assessing short summaries. Our proposed method integrates the semantic relations between words, and their syntactic composition. As a result, the proposed method is able to obtain high accuracy and improve the performance compared with the current techniques. Experiments have displayed that it is to be preferred over the existing techniques. A summary evaluation system based on the proposed method has also been developed.

1. Introduction

Summarization is the process of automatically producing a compressed version of a given text that provides useful information for the user (Aliguliyev, 2009; Zipitria, Larrañaga, Armañanzas, Arruarte, & Elorriaga, 2008). Summary writing is an important part of many English Language Examinations (He, Hui, & Quan, 2009). Summarizing instructs students concerning how to recognize the main ideas in a text, how to determine important information that is worth noting and how to eliminate irrelevant information (Brown & Day, 1983; Chang, Sung, & Chen, 2002; Zipitria, Arruarte, & Elorriaga, 2010; Zipitria, Elorriaga, Arruarte, & de Ilarraza, 2004). It can also be useful for the instruction of second language students (Alyousef, 2006; Cho, 2012; Fan, 2010; Hedge, 2001; Pakzadian & Rasekh, 2013).

Since summarization can be used as a measure of understanding for a given text, teachers need to grade students' written summaries. If they do it manually, it is a difficult and very time-consuming task. In order to reduce the time they should spend on this task, many teachers have decided to reduce the number of summaries given to their students. However, if students do not have sufficient practice it affects their summary writing ability. To tackle this problem, computer-assisted

* Corresponding author.
E-mail addresses: asadabdi55@gmail.com (A. Abdi), norisma@um.edu.my (N. Idris), r.alguliev@gmail.com (R.M. Alguliev), r.aliguliyev@gmail.com (R.M. Aliguliyev).

http://dx.doi.org/10.1016/j.ipm.2015.02.001
0306-4573/

assessment (CAA), which has attracted interest in recent years, can help teachers. Due to the progress in other areas, such as E-learning, Information Extraction and Natural Language Processing, automatic evaluation of summaries has been made possible. Several techniques such as Latent Semantic Analysis (LSA) (Franzke & Streeter, 2006; Landauer, Laham, Rehder, & Schreiner, 1997; Zipitria et al., 2004), ROUGE (Lin, 2004), BLEU (Pérez, Alfonseca, & Rodrıguez, 2004) and N-gram Co-occurrence (Lin, 2004) have been proposed for the automatic assessment of summaries. Summary writing assessment can be divided into content and linguistic quality (Jones & Galliers, 1996; Valenti, Neri, & Cucchiarelli, 2003). Whereas content assessment determines how much the information of the source text overlaps with the information in the summary text, linguistic quality assessment judges the accurate spelling and grammar of summaries, etc. In this work, we focused on the content evaluation.

Most of the existing assessment methods have focused on long texts, in which they calculate the similarity between the long texts based on the shared words. These methods are suitable for long texts because two similar long texts usually have a number of co-occurring words. However, a short text may only contain a few words co-occurrence or even none. This is because people can express the same meaning using various sentences in terms of word content. In addition, some of the existing methods do not contribute syntactic information to calculate the similarity between texts. However, for the correct assessment of summaries, a method should consider both semantic and syntactic information in evaluation (Kanejiya, Kumar, & Prasad, 2003; Pérez et al., 2005; Wiemer-Hastings & Wiemer, 2000; Wiemer-Hastings & Zipitria, 2001).

In this paper, we propose a method that merges semantic relations between words, and their syntactic composition for improving the accuracy of automatic summary evaluation. The semantic similarity is computed using information from a lexical database. The use of a lexical database enables our method to model human common sense knowledge.

The proposed method is called SALK: summarization assessment based on linguistic knowledge. SALK can be applied to both short text and long text. The proposed method can be employed in various applications in areas such as text mining (Atkinson-Abutridy, Mellish, & Aitken, 2004), text summarization (Erkan & Radev, 2004), text classification (Ko, Park, & Seo, 2004) and education (Foltz, Laham, & Landauer, 1999; Franzke & Streeter, 2006).

The rest of the paper is structured as follows. Section 2 reviews some of the proposed techniques that are used to evaluate summaries. Section 3 introduces the proposed method. Section 4 presents the developed automatic summary evaluation system. Section 5 discusses the performance analysis and presents the results of the analysis. Finally, in section 6, we summarize the works discussed and the progress of the project.

2. Related work

In the recent past, summary assessment has become one of the investigated topics in natural language processing. Several studies have shown that computer can be used for summary assessment. Thus, several techniques have proposed to assess summaries. In this section detailed information related to assessment approaches of the summaries are given.

LSA (Landauer, 2002) is a statistical technique for the representation of the meaning of words and sentences. It has been used in educational applications, such as essay grading (Landauer & Dumais, 1997), as well as in NLP applications containing information retrieval (Landauer et al., 1997) and text segmentation (Choi, Wiemer-Hastings, & Moore, 2001). It only requires raw text as its input. In the first step, it represents the text as a matrix in which each row represents a unique word and each column represents a text passage or sentence. Each cell is used to represent the importance of words in sentences. Different approaches can be used to fill out the cell values, such as frequency of words, Binary Representation and Term Frequency-Inverse Document Frequency. The next step is applying Singular Value Decomposition (SVD) to the matrix. This is a form of factor analysis. LSA has some disadvantages: the first is that it does not use syntactic composition, such as word order, which this information is necessary to understand the meaning of two sentences. The second is that it can produce a reasonable result when it takes a large corpus as its inputs. The third limitation is that since all words do not appear in all sentences, the matrix created is usually sparse.

Laburpen Ebaluaka Automatikoa (LEA) (Zipitria et al., 2004), which is based on Latent Semantic Analysis (LSA), has been proposed to evaluate the summary. It is designed for both teachers and students. It allows teachers to examine a student's written summary, and allows students to produce a summary text with their own words. The summaries are evaluated based on certain features, such as cohesion, coherence, use of language and the adequacy of the summary.

Summary Street (Franzke & Streeter, 2006), which is based on LSA is a computer-based assessment system that is used to evaluate the content of the summary text. Summary Street ranks a student's written summary by comparing the summary text and the source text. It creates an environment to give appropriate feedback to the students, such as content coverage, length, redundancy and plagiarism.

Lin (2004) proposed an automatic summary assessment system named Recall-Oriented Understudy for Gisting Evaluation, which is used to assess the quality of the summary text. The current system includes various automatic assessment approaches, such as ROUGE-N, ROUGE-L, and ROUGE-S. ROUGE-N is an n-gram recall between a candidate summary and a reference summary. ROUGE-L calculates the similarity between a reference summary and a candidate summary based on the Longest Common Subsequence (LCS). ROUGE-S is a measure of the overlap of skip-bigrams between a candidate and a reference summary.

Mohler, Bunescu, and Mihalcea (2011) introduced an Answer Grading System, which combines a graph alignment model and a text similarity model. This system aims to improve the existing approaches that automatically assign a grade to an

answer provided by a student, using the dependency parse structure of a text and machine learning techniques. The current system uses the Stanford Dependency Parser (De Marneffe, MacCartney, & Manning, 2006) to create the dependency graphs for both the student (A_1) and teacher (A_2) answers. For each node in the student's dependency graph the system computes a similarity score for each node in the teacher's dependency graph using a set of lexical, semantic, and syntactic features. The similarity scores are used to weight the edges that connect the nodes in A_1 on one side and the nodes in A_2 on the other. The system then applies the Hungarian algorithm to determine both an optimal matching and the score associated with such a matching for the answer pair. Finally, the system produces a total grade based on the alignment scores and semantic similarity measures.

Rus, Graesser, and Desai (2007) proposed a graph-based method to recognize textual entailment (RTE). The RTE is based on the idea of subsumption. A text T subsumes a text H if text T is more general than or identical to text H. This idea is applied to textual entailment. However, text T entails text H if and only if text T subsumes text H. The graph representation has been used for the text T and text H to implement the idea of subsumption. Given two texts, T and H, the method maps those into two graph-structures, one for H and one for T, where nodes represent concepts and edges represent lexico-syntactic relations among the concepts. Based on the subsumption score between the T-graph and H-graph, the method can make a decision concerning whether the meaning of text T can be derived from text H.

Kanejiya et al. (2003) proposed an approach called Syntactically Enhanced LSA (SELSA) for the assessment of students' answers. This approach is similar to tagged LSA (Wiemer-Hastings & Zipitria, 2001). SELSA considers both semantics and syntax in the answers. It improves LSA by considering a word along with the part-of-speech (POS) tag of its preceding word. In this approach, LSA consists of a matrix in which each row corresponds to the combination of word-prevtag pairs and the column corresponds to the document. Prevtag indicates the POS tag of the preceding word and the document can be sentence, paragraph or a larger unit of text.

Wiemer-Hastings and Zipitria (2001) proposed a model called the tagged model or tagged LSA, which adds syntactic information to the LSA for the comparison of two texts. It extracts the structural information from the text using part-of-speech tagging. Brill's tagger (1994) has been used to assign a unique part-of-speech (POS) tag to every word in the text. LSA would consider each word/tag as a single term. When LSA does not distinguish between words that are used in different parts of speech, it would be able to distinguish the POS for each word in comparing texts. The experiment results have shown that the performance of the tagged LSA was not satisfactory. However, in another effort Wiemer-Hastings and Zipitria (2001) proposed an approach called structured LSA (SLSA). SLSA segments sentences into several substrings, such as noun phrase, verb phrase and object phrase, and then it calculates the similarity between two sentences using an averaging of the LSA based similarity of noun phrase, verb phrase and object phrase. The results of the experiment have shown that the approach could improve the performance in terms of sentence-pair similarity judgment and the structural information.

3. Proposed method: SALK

Fig. 1 displays the general architecture of our proposed method. The method can be used to assess short summaries. It is called Summarization Assessment based on Linguistic Knowledge (SALK), since the summaries are evaluated using semantic information obtained from a lexical database and syntactic information is given by analyzing the structure of the sentence. The proposed method contains the following steps:

i. *Pre-processing.* The goal of this stage is to prepare a summary text and original text for the subsequent stages.
ii. *Intermediate-processing.* In this stage, the measure of similarity between each sentence of the summary text and a set of sentences from the original text is computed, and the highest similarity measure is assigned to the current sentence from the summary text.
iii. *Post-processing.* This aims to calculate the final similarity score using Eqs. (6) and (7).

The tasks of each stage are as follows:

3.1. Pre-processing

This stage aims to perform a basic linguistic analysis on both the source text and summary text. Thus, it prepares them for further processing. The pre-processing module provides text pre-processing functions, such as sentence segmentation, tokenization and stop word removal.

3.2. Intermediate-processing

Intermediate processing is the core of the proposed method to assess the summaries. Fig. 2 shows the overall process of applying the semantic and syntactic information to summary evaluation. First, the source text and the summary text are decomposed into a set of sentences. Then, the similarity measure between each sentence from the summary text and whole sentences from the source text is determined using the composition of word order similarity and semantic similarity. The maximum value is assigned as a similarity score for the current sentence of the summary. Fig. 2 includes a few components

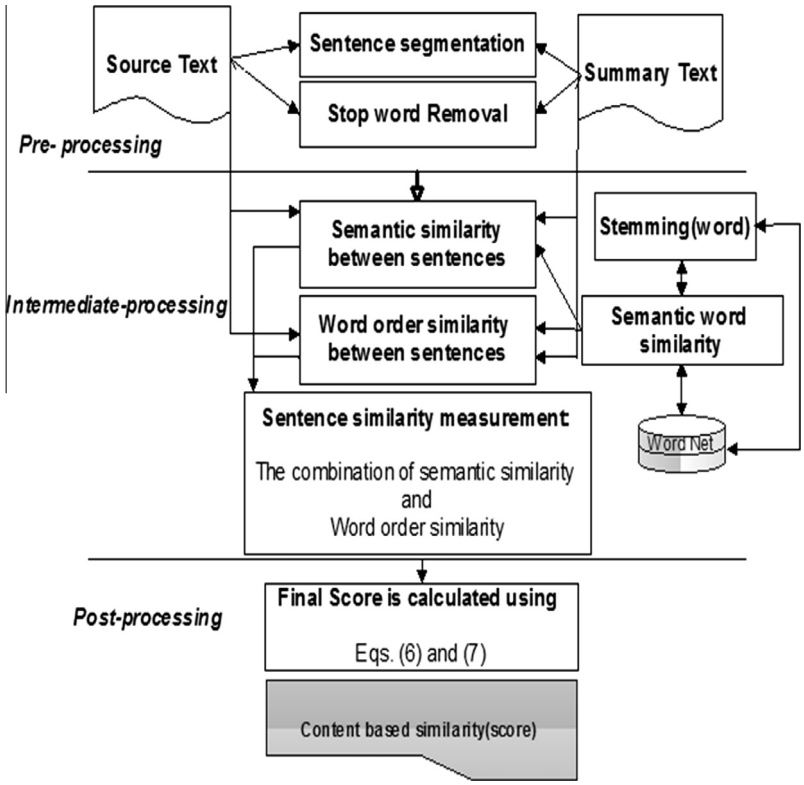

Fig. 1. The architecture of the SALK.

such as word set, semantic similarity between words, semantic similarity and syntactic similarity between sentences. The tasks of each component are as follows:

3.2.1. The word set

Given two sentences S_1 and S_2, a "word Set" is created using distinct words from the pair of sentences. Let $WS = \{W_1, W_2, \ldots, W_N\}$ denote word set, where N is the number of distinct words in word set. The word set between two sentences is obtained through certain steps. At first, it takes two sentences as its input. In the second step, by a loop for each word W from S_1, it undertakes certain tasks, which include determining the root of the current word W using the WordNet. Then, it checks if the word appears in the *WS*, it jumps to the second step and continues the loop by the next word from S_1, otherwise, if the word does not appear in the *WS*, the word W is assigned to the *WS* and then it jumps to the second step to continue the loop by the next word from S_1 . It conducts the same process for Sentence 2. The corresponding process is shown in algorithm 1.

Algorithm 1. The creation of "word set"

Input: Sentence 1, Sentence 2;
Output: $WS = \{W_1, W_2, \ldots, W_n\}$, *WS* denotes an array that includes all distinct words from two sentences;
1: Let W be a word of the Sentence 1 or Sentence 2;
2: Let *RW* be the root of word W, it is obtained using Word Net;
3: Let L be the length of Sentence 1 or Sentence 2;
4: Set $l = 0$;
5: For each W,
  i. l = l + 1;
  ii. Get *RW*;
  iii. Look for *RW* in word set;
  iv. If the *RW* was not in *WS*, then assign *RW* to *WS*; otherwise, jump to step 6;
6: Jump to step 5; iterate until $l \leqslant L$;

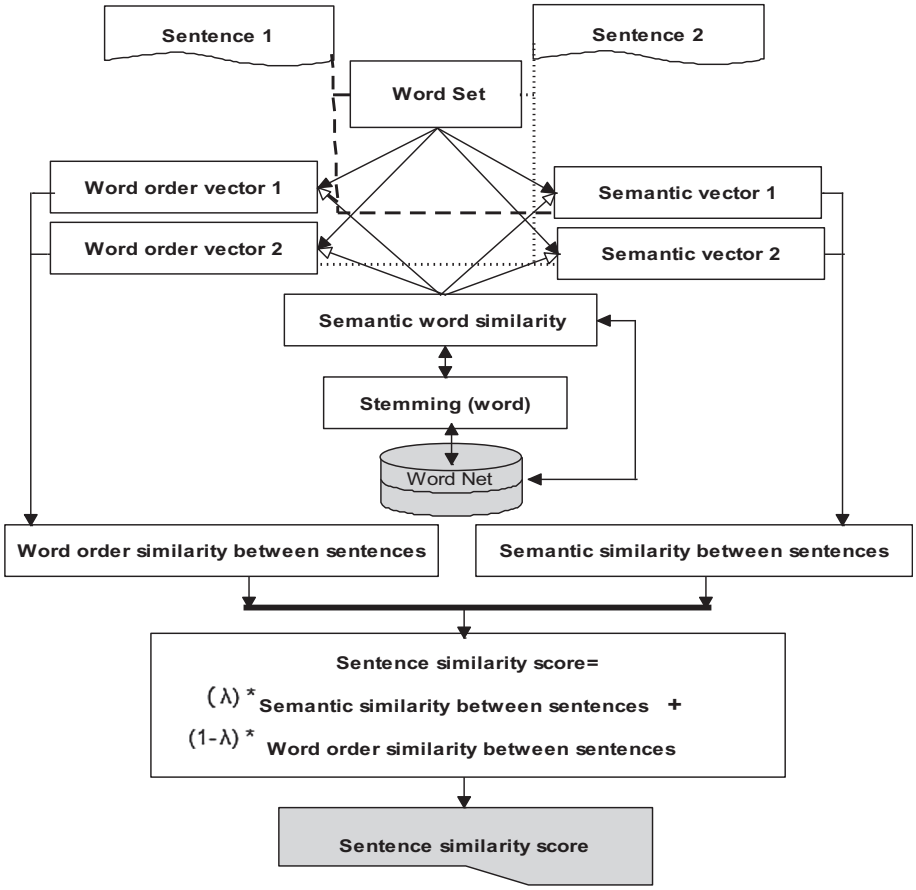

Fig. 2. Sentence similarity computation.

3.2.2. Semantic similarity between words

Semantic word similarity (Lin, 1998; Tian, Li, Cai, & Zhao, 2010) plays an important role in this method. It is used to create a word order vector and semantic vector. Given two words W_1 and W_2, the semantic similarity between words is determined through these steps:

I. Get root of each word using the lexical database, Word Net;
II. Get synonym of each word using the lexical database, Word Net;
III. Determine the number of synonyms of each word;
IV. Determine Least Common Subsume (LCS) of two words and their length;
V. Calculate the similarity score between words using Eqs. (1) and (2);

We use the following equations to calculate the semantic similarity between words (Aytar, Shah, & Luo, 2008; Lin, 1998; Mihalcea, Corley, & Strapparava, 2006; Warin, 2004):

$$IC(w) = 1 - \frac{Log(synset(w) + 1)}{\log(\max\_w)} \tag{1}$$

$$Sim(w_1, w_2) = \begin{cases} \frac{2*IC(LCS(w_1,w_2))}{IC(w_1)+IC(w_2)} & \text{if } w_1 \neq w_2 \\ 1 & \text{if } w_1 = w_2 \end{cases} \tag{2}$$

where *LCS* stands for the least common subsume, max_*w* is the number of words in Word Net, Synset (w) is the number of synonyms of word w, and $IC(w)$ is the information content of word *w* based on the lexical database Word Net.

3.2.3. Semantic similarity between sentences

We use the semantic-vector approach (Alguliev, Aliguliyev, & Mehdiyev, 2011; Aliguliyev, 2009; Li, McLean, Bandar, O'shea, & Crockett, 2006) to measure the semantic similarity between sentences. The semantic vector is derived from the

word set and corresponding sentence. Each cell of the semantic vector corresponds to a word in the word set, so the dimension equals the number of words in the word set. Each cell of the semantic vector is weighted using the calculated semantic similarity between words from the word set and corresponding sentence. As an example, if the word W from the word set appears in the sentence S_1, the weight of word W in the semantic vector is set to 1. Otherwise, if the word W does not appear in the sentence S_1, then we use the word similarity method introduced in Section 3.2.2 to calculate the similarity between the word W and all of the words in the sentence S_1. This is done to find the most similar ones. Then, the weight of word W in the semantic vector is set to the highest similarity value. If there is no similarity value between word W and all of the words in the sentence S_1, the weight of word W in the semantic vector is set to 0. A semantic-vector is created for each of the two sentences. The corresponding process is shown in algorithm 2. The semantic similarity measure is computed based on the two semantic vectors. The following equation is used to calculate the semantic similarity between sentences:

$$Sim_{semantic}(S_1, S_2) = \frac{\sum_{j=1}^{m}(w_{1j} \times w_{2j})}{\sqrt{\sum_{j=1}^{m} w_{1j}^2} \times \sqrt{\sum_{j=1}^{m} w_{2j}^2}} \tag{3}$$

where $\boldsymbol{S}_1 = (w_{11}, w_{12}, \ldots, w_{1m})$ and $\boldsymbol{S}_2 = (w_{21}, w_{22}, \ldots, w_{2m})$ are the semantic vectors of sentences S_1 and S_2, respectively; w_{pj} is the weight of the j^{th} word in vector $\boldsymbol{S}_p$, m is the number of words.

Algorithm 2. Lexical semantic vector

Input: sentence 1, sentence 2, "word set";
Output: semantic vector;
1: Let S be either sentence 1 or sentence 2;
2: Let W_t be a word of the word set;
3: Let RW be the root of the word W_t, it is obtained using the Word Net;
4: Let W be a word of S;
5: Let SSM denotes the semantic similarity measure between words;
6: Let L be the length of S;
7: Set $l = 0$;
8: For each W_t,
  i. l = l + 1;
  ii. Get RW;
  iii. Look for RW in S;
  iv. If RW was in S, then set corresponding element in semantic vector to "1";
  v. Otherwise,
    a. For each W,
      1. $SSM(W, W_t)$ is calculated using Eqs. (1) and (2);
      2. If $SSM > 0$, Then assign SSM to array1;
      3. Iterate until $l \leqslant L$;
    b. If array1 = Null, then jump to step 9; otherwise,
    c. Select the most similarity value from array1;
    d. Set the corresponding element of the vector to the most value of similarity measure; set $l = 0$; jump to step 8;
  9: Assign "**0**" to the corresponding element of the vector; jump to step 8; iterate until $l \leqslant L$;

3.2.4. Word order similarity between sentences

We use the syntactic-vector approach (Li et al., 2006) to measure the word order similarity between sentences. The syntactic-vector is derived from the word set and corresponding sentence, so the dimension is equal to the number of words in the word set. Unlike the semantic-vector, each cell of the syntactic-vector is weighted using a unique index. The unique index can be the index position of the words that appear in the corresponding sentence. To create a syntactic-vector we follow these steps, for each word w from the word set. If the W appears in the sentence S_1 the cell in the syntactic-vector is set to the index position of the corresponding word in the sentence S_1. Otherwise, if the word W does not appear in the sentence S_1, a semantic similarity measure is calculated between word W and each word from the sentence S_1 using Eqs. (1) and (2). Finally, the value of the cell is set to the index position of the word from the sentence S_1 with the highest similarity measure. If there is not a similar value between the word W and all of the words in the sentence S_1, the weight of the cell in the syntactic-vector is set to 0. A syntactic-vector is created for each of the two sentences. The Algorithm 3 presents the steps to create the word order vector. The syntactic similarity measure is computed based on the two syntactic-vectors. The following equation is used to calculate word order similarity between sentences:

$$\mathrm{Sim}_{\text{word order}}(\boldsymbol{S}_1, \boldsymbol{S}_2) = 1 - \frac{||O_1 - O_2||}{||O_1 + O_2||} \tag{4}$$

where $\boldsymbol{O}_1 = (d_{11}, d_{12}, \ldots, d_{1m})$ and $\boldsymbol{O}_2 = (d_{21}, d_{22}, \ldots, d_{2m})$ are the syntactic vectors of sentences S_1 and S_2, respectively; d_{pj} is the weight of the j^{th} cell in vector $\boldsymbol{O}_p$.

Algorithm 3. Word order vector

Input: sentence 1, sentence 2, "word set";
Output: Lexical vector;
1: Let S be either sentence 1 or sentence 2;
2: Let W_t be a word of the word set;
3: Let RW be the root of the word W_t, it is obtained using the Word Net;
4: Let W be a word of S;
5: Let SSM denotes the semantic similarity measure;
6: Let L be the length of S;
7: Set l =0;
8: For each W_t,
  i. $l = l + 1$;
  ii. Get RW;
  iii. Look for RW in S;
  iv. If RW was in Sen, then set corresponding element in vector to index position of word in S;
  v. Otherwise,
    a. For each W,
      1. $SSM(W, W_t)$ is calculated using Eqs. (1) and (2);
      2. If $SSM > 0$, *Then assign SSM to array1*;
      3. *Iterate until* $l \leqslant L$;
    b. If array1 = Null, then jump to step 9; otherwise,
    c. Select the most similarity score from array1;
    d. Set the corresponding element of vector to index position of word with the most similarity score; set $l = 0$; jump to step8;
  9: Assign '**0**″ to the corresponding element of the vector; jump to step 8; iterate until $l \leqslant L$;

3.2.5. Sentence similarity measurement

We calculate the similarity measure between two sentences using a linear equation that combines the semantic and word order similarity. The similarity measure is computed as follows:

$$\mathrm{sim}_{\mathrm{sentences}}(S_1, S_2) = \lambda \cdot \mathrm{sim}_{\mathrm{semantic}}(S_1, S_2) + (1 - \lambda) \cdot \mathrm{sim}_{\mathrm{wordorder}}(S_1, S_2) \tag{5}$$

where $0 \ll 1$ is the weighting parameter, specifying the relative contributions to the overall similarity measure from the semantic and syntactic similarity measures. The larger the, heavier the weight for the semantic similarity. If = 0.5 the semantic and syntactic similarity measures are assumed to be equally important.

3.3. Post-processing

This displays the results from the system to the user. It shows the similarity measure as a score to the user. We use the following equations to calculate the Final Score (FS) for any student's written summary:

$$FS = (\mathrm{TBP} \times \frac{\sum_{S \,\epsilon\, S_{\mathrm{summary}}} \mathrm{MSS}(S)}{N}) \times 100 \tag{6}$$

$$TBP = \begin{cases} 1 & if\ |N_s| > |N_t| \\ e^{\left(1 - \frac{N_t}{N_s}\right)} & if\ |N_s| \leqslant |N_t| \end{cases} \tag{7}$$

where $\mathrm{S}_{\mathrm{summary}} = \{S_1, S_2 \cdots S_N\}$ includes all the sentences in the summary text, where N is the total number of sentences in the summary text. *MSS* is the Maximum Similarity Score between a sentence from the summary text and all the sentences from the source text. $|N_s|$ and $|N_t|$ are the total number of sentences in the summary text and the source text, respectively. The Text Brevity Penalty (TBP) is calculated to prevent very short summary texts that try to increase their similarity scores.

4. System implementation

An automatic summarization assessment system based on our method, which contains six main modules, is displayed in Fig. 3.

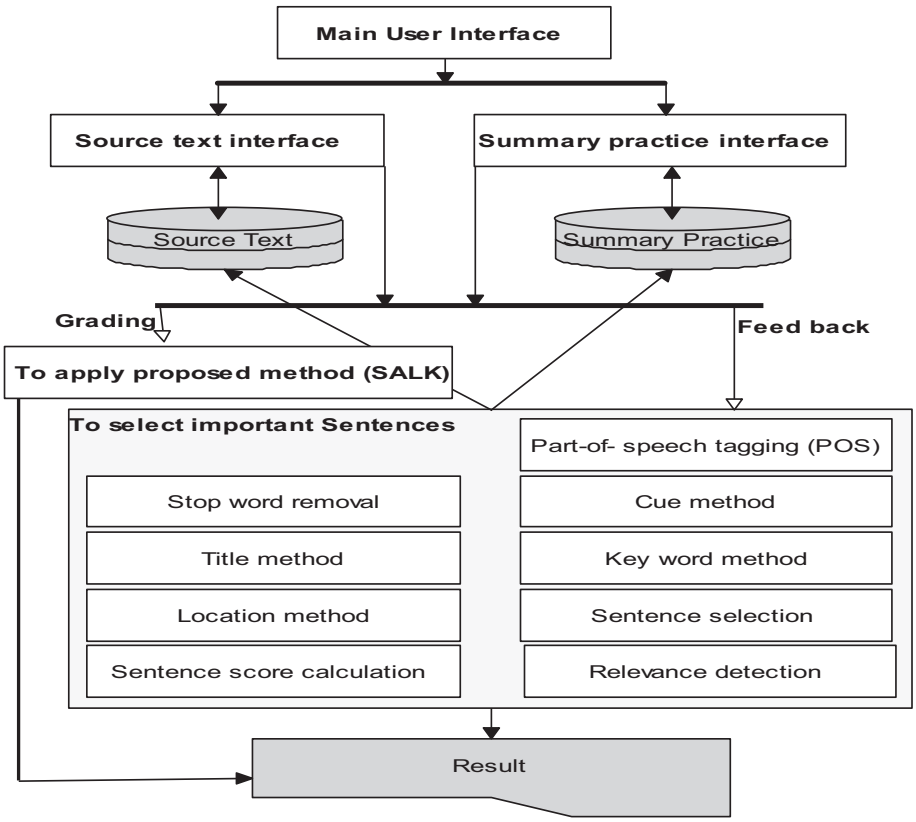

Fig. 3. The summary assessment system.

- *Main User Interface* – allows the user to switch between the source text interface and the summary practice interface.
- *Source text interface* – allows the user to upload the source text for summary exercise.
- *Summary practice interface* – allows the user to exercise their summary writing ability using the source text.
- *Grading* – aims to apply the proposed method to the student summary to determine the similarity measure between the summary text and the original text.
- *Feedback* – provides appropriate feedback to students about the content of the summary text, which includes the module of important sentences.
- *Important sentences* – the aim of this module is to determine important sentences from the original text.
- *Result interface* – it displays the similarity measure as a score and an appropriate feedback to the user. It also shows which sentences from the source text have been selected to create a summary text.

To give appropriate feedback to students about the missing information in each student summary, the module of important sentences identifies the most important sentences using a linear method (Edmundson, 1969). Edmundson (1969) proposed a method that considers the title word, cue word, keyword and sentence location for identifying important sentences in the original text. When a student submits a summary, each sentence of the summary text is compared to the important sentences. If a matched sentence does not exist in the summary text, the system provides feedback in the form of a comment to the student. However, it is suggested students, "you can also select these sentences". In this case, the students can revise their summaries and resubmit them to the system.

The module of important sentences consists of a few processes:

Part-of- speech tagging – is an external tool that assigns each word its morphological category, such as noun, verb or adjective. The results of this function are sent to the keyword method and title method.
Key method (Alonso et al., 2004) – uses the term frequency (*TF*) method to identify words with high frequency. In this work, five words with the most frequency are selected as keywords.
Location method (Fattah & Ren, 2009; Kupiec, Pedersen, & Chen, 1995) – finds the location of each sentence in the original text and determines whether it is the first or the last sentence of a paragraph.
Title method (Kupiec et al., 1995) – extracts all the nouns and verbs from the title of the original text.

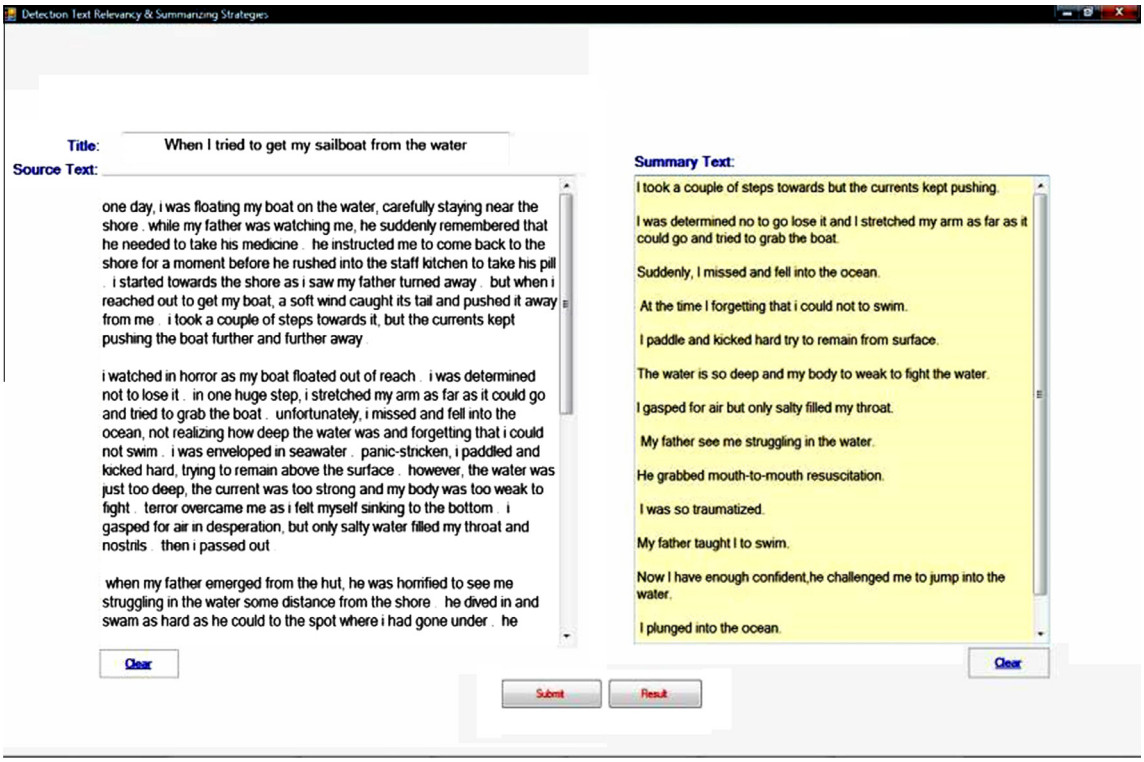

Fig. 4. The main interface of system.

Cue method – cue words, such as "consequently" or "in particular", are often followed by important information. Thus, sentences that contain one or more of these cue phrases are considered more important than sentences without cue phrases (Zhang, Sun, & Zhou, 2005).
Sentence score calculation – the sentence score is calculated as follows. If a sentence includes a cue word, then it assigns a score to the sentence (denoted by C). If a sentence is the first or the last sentence of a paragraph, then it assigns a score to the sentence (denoted by L). If a sentence includes a word of the title words, then it assigns a score to the sentence (denoted by T). If a sentence includes a word of the key words, then it assigns a score to the sentence (denoted by K).

$$Score_{sentence} = C + L + T + K \tag{8}$$

where C, K, L, T are fixed values of cue words, key words, sentence location and title words, respectively.
Sentence selection – chooses N sentences with a high score (N is a predefined value).
Relevance detection – each sentence of summary text is compared to the important sentences. If it does not exist in a matched sentence in summary text, the system provides feedback to the student.

This module also uses the lexical database for English, Word Net, developed at Princeton University (Miller & Charles, 1991). It includes 121,962 unique words, 99,642 synsets (each synset is a lexical concept represented by a set of synonymous words) and 173,941 senses of words.

Fig. 4 shows the main interface. It allows the user to switch between the source text interface and the summary exercise interface. The user can upload the source text and practice summary writing. After the user has produced a summary, he/she can assess the summary by clicking on the submit button. In this case, the system applies the proposed method and the module of important sentences to the summary text and original text respectively. After the user's summary is assessed by the system, the results will be displayed to the user using the result interface component, as shown in Fig. 5.

5. Experiments

In this section, to examine the efficiency of the SALK, the performance of the SALK is compared with the existing assessment techniques, such as LSA, N-gram, BLEU, LSA_Ngram, LSA_ERB and ROUGE (N-gram, LCS, Skip bigram), which are employed by most of the important summary evaluation systems. To do this, we now explain our experiments on the single-document summarization datasets provided by Document Understanding Conference (http://duc.nist.gov).

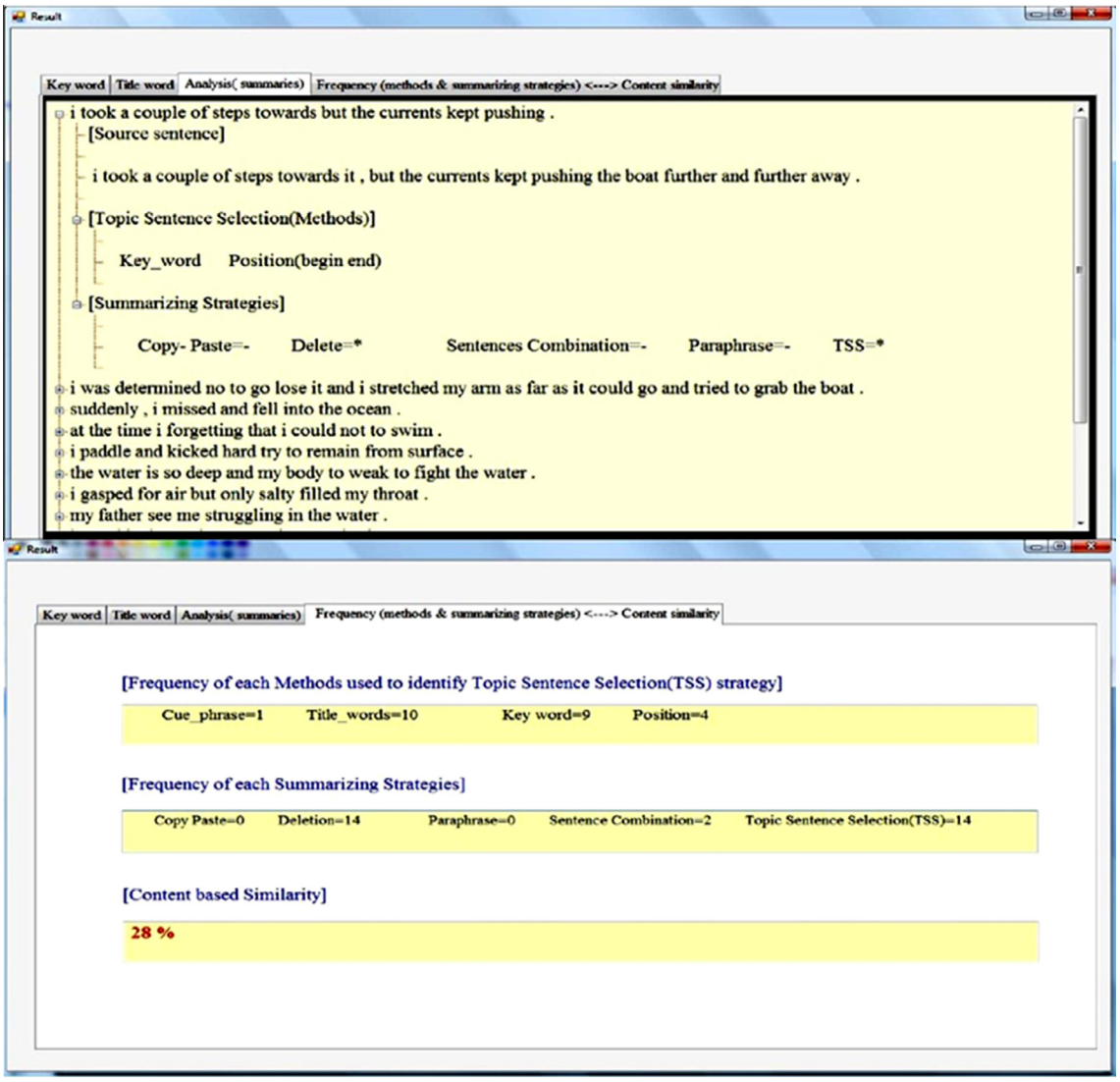

Fig. 5. The result interface of system.

5.1. Data set

In this section, we describe the data used throughout our experiments. For assessment the performance of the proposed method we used the document datasets DUC 2002 and corresponding 100-word summaries generated for each of documents. DUC 2002 contains 567 documents-summary pairs from Document Understanding Conference. It is worth mentioning that each document of DUC 2002 is denoted by original text or source text and the corresponding summary is denoted by candidate summary. In our experiments, the documents and corresponding summaries were randomly divided into two separate dataset. In the first experiment, 300 documents and corresponding summaries were used for parameter tuning

Table 1
Description of dataset.

	DUC 2002
Number of cluster	59
Number of documents in each cluster	~10
Number of documents	567
Data source	TREC
Summary length	100 words

(the threshold and lambda). In the second experiment, the performance of the proposed method was compared with the existing assessment techniques using the remaining documents and corresponding summaries. Table 1 gives a brief description of the datasets.

Participants – four experts, (1) an English teacher with good reading skills and understanding ability in the English language as well as experience in teaching summary writing; (2) a lecturer with experience in using the skills in their teaching method; and (3) two of them were doctorate candidates with good reading skills and understanding ability in the English language participated in the current study.
Human expert assessment – the human experts were asked to score the candidate summaries on a scale of 0–1 with respect to how much the information in the original text overlapped with the information in the student summary. Each of the two assessors independently produced scores for every summary. Finally, each of the candidate summaries was assigned the average of the scores that the human experts had assigned to each student summary.
Inter-raters agreement – we used Pearson correlation coefficient as a measure of agreement between the four raters. The average correlation coefficient between the human raters was 0.61. This value indicated that our assessors had good agreement (Landis & Koch, 1977) for grading each candidate summary.

Moreover, in order to conduct the similar test, dissimilar test and synonym test, the human experts were asked to create reference similar (denoted by Ref_{sim}), reference dissimilar (denoted by Ref_{diss}) and reference synonym (denoted by Ref_{syn}) for each candidate summary. These references were produced using the following procedure. Each human expert was given 141 summaries. The human expert would break every candidate summary into a number of sentences and then compare each sentence of the candidate summary with all sentences from the original text irrespective of whether or not two sentences were semantically identical. Semantically identical sentences include the same information or talk about a similar idea. However, the group of sentences from the original text that were semantically equivalent with sentences in the candidate summary could be considered as the reference similar. The remaining sentences from the original text that were not semantically equivalent with the sentences of the candidate summary were considered as the reference dissimilar. In order to make a common basis for comparing Ref_{sim} and candidate summary, if the candidate summary used different words from the Ref_{sim} (i.e., a word of reference similar replaced by a synonym or similar word in the candidate summary), all the words were converted to a common word. In the synonym test, unlike the similar test, for comparing Ref_{sim} and the candidate summary, if the candidate summary used different words from the Ref_{sim}, these words were not converted to a common one.

5.2. Assessment techniques

In order to make inferences about our proposed method, we compare the SALK with the other existing base assessment techniques, such as LSA, N-gram, BLEU, LSA_Ngram, LSA_ERB and ROUGE (N-gram, LCS, Skip bigram). To this end, in the first steps we present each of these assessment techniques.

An n-gram is a sequence of n items from a text. In our experience, an item refers to the word. An n-gram of size 1, 2 and 3 is referred to as "unigram", "bigram" and "trigram", respectively. The large size is called "n-gram" (e.g., six-gram). This measures the similarity between two texts based on the number of n-grams co-occurring in two compared texts. The N-gram match ratio is calculated as follows:

$$C_n = \frac{\sum_{S \in \{\text{Reference summaries}\}} \sum_{\text{N-gram} \in S} \text{Count}_{\text{match}}(\text{N-gram})}{\sum_{S \in \{\text{Reference summaries}\}} \sum_{\text{N-gram} \in S} \text{Count}(\text{N-gram})} \tag{9}$$

where N is used for the length of the N-gram and $\text{Count}_{\text{match}}$ (N-gram) is the total number of N-grams co-occurring in a reference summary and a candidate summary. Count (N-gram) is the number of N-grams in the reference summaries. The N-gram match ratio (Eq. (9)) is used to compare all the sentences in the reference summary and candidate summaries as follows:

$$\text{N-gram}(m, n) = e^{\left(\sum_{n=1}^{m} W_n \times \log C_n\right)} \tag{10}$$

where $m \geqslant n$, n and m range from 1 to 4, $W_n = 1/(m - n + 1)$.

ROUGE metric is also used to evaluate summaries. It contains ROUGE-N, ROUGE-L and ROUGE-S that compute the similarity between two texts. We describe these methods as follows.

ROUGE-N, compares two summaries based on total number of matches. It is calculated as follows:

$$\text{ROUGE} - \text{N} = \frac{\sum_{S \in \{\text{Reference summaries}\}} \sum_{\text{N-gram} \in S} \text{Count}_{\text{match}}(\text{N-gram})}{\sum_{S \in \{\text{Reference summaries}\}} \sum_{\text{N-gram} \in S} \text{Count}(\text{N-gram})} \tag{11}$$

where N is used for the length of the N-gram and $\text{Count}_{\text{match}}$ (N-gram) is the total number of N-grams co-occurring in a reference summary and a candidate summary. Count (N-gram) is the number of N-grams in the reference summaries.

ROUGE-L calculates the similarity between a reference summary and a candidate summary based on the Longest Common Subsequence (LCS). It measures the similarity between two summaries using Eq. (12).

$$\begin{cases} P_{\text{lcs}} = \frac{LCS(R,S)}{N} \\ R_{\text{lcs}} = \frac{LCS(R,S)}{M} \\ F_{\text{lcs}} = \frac{(1+\beta^2)R_{lcs}\times P_{lcs}}{\beta^2\times R_{lcs}\times P_{lcs}} \end{cases} \tag{12}$$

where R is a reference summary and S is a candidate summary. M is the length of reference summary and N is the length of the candidate summary. $LCS(R,S)$ is the length of an LCS of the reference summary and the candidate summary. $P_{lcs}(R,S)$ computes the precision of $LCS(R,S)$, $R_{lcs}(R,S)$ computes the recall of $LCS(R,S)$ and $\beta = P_{lcs}(R,S)/R_{lcs}(R,S)$.

ROUGE-S (Skip-Bigram Co-Occurrence), skip-bigram is any pair of words in their sentence order, allowing for arbitrary gaps. ROUGE-S measures the similarity between two summaries based on common skip-bigrams using Eq. (13).

$$\begin{cases} P_{\text{skip2}} = \frac{SKIP2(R,S)}{C(N,2)} \\ R_{\text{skip2}} = \frac{SKIP2(R,S)}{C(M,2)} \\ F_{\text{skip2}} = \frac{\left(1+\beta^2\right)R_{skip2}\times P_{skip2}}{\beta^2\times R_{skip2}\times P_{skip2}} \end{cases} \tag{13}$$

where R is a reference summary and S is a candidate summary. M is the length of the reference summary and N is the length of the candidate summary. $SKIP2(R,S)$ is the total number of skip-bigram matches between two summaries, reference summaries and candidate summaries. $P_{skip2}(R,S)$ computes the precision of $SKIP2(R,S)$, $R_{skip2}(R,S)$ computes the recall of $SKIP2(R,S)$ and $\beta = P_{skip2}(R,S)/R_{skip2}(R,S)$. $C(n,2)$ and $C(m,2)$ are the combination functions.

LSA in the first step represents the text as a matrix in which each row represents a unique word and each column represents a text passage or sentence. If there are m distinct words and n sentences, the matrix X is created with the size of $m \times n$. Each cell is used to represent the importance of the words in the sentences. Different approaches can be used to fill out the cell values. These approaches are as follows:

- *Term frequency*: the cell is filled out with the frequency of the word in the sentence.
- *Binary Value*: the cell is filled out with 0/1 according to the existence of the word in the sentence.
- *Term Frequency–Inverse Sentence Frequency*: the cell is filled using the $Tf - ISf$ value of the term. The term frequency (TF) value is the number of occurrences of the term in a sentence. The inverse sentence frequency (ISF) value is calculated using Eq. (14):

$$\text{ISF} = \log\frac{|N|}{n_i} \tag{14}$$

where $|N|$ is the total number of sentences in the input text, and n_i is the number of sentences that contain the jth word.

LSA applies singular value decomposition (SVD) to the matrix X. SVD can be seen as a method for data reduction. A rectangular matrix $\boldsymbol{X}_{m\times n}$ can break down into the product of three matrices: an orthogonal matrix $\boldsymbol{U}_{m\times n}$, which is called the left singular vector. A diagonal matrix $\boldsymbol{K}_{n\times n}$, which is called the singular value. The transpose of an orthogonal matrix $\boldsymbol{V}_{m\times n}$, which is called the right singular vector. The equation for singular value decomposition of $\boldsymbol{X}$ is as follows:

$$\boldsymbol{X}_{m\times n} = \boldsymbol{U}_{m\times n}\times\boldsymbol{K}_{n\times n}\times\boldsymbol{V}^T_{n\times n} \tag{15}$$

Finally, in order to reduce the number of dimensions, a few elements of matrix $\boldsymbol{K}_{n\times n}$ are deleted, and the result of matrix is $\boldsymbol{K}'_{1\times 1}$. A new matrix, $\boldsymbol{X}'_{m\times 1}$, is created by multiplying three matrixes. $\boldsymbol{X}'_{m\times 1}$ is defined as follows:

$$\boldsymbol{X}'_{m\times 1} = \boldsymbol{U}_{m\times 1}\times\boldsymbol{K}'_{1\times 1}\times\boldsymbol{V}'^T_{1\times 1} \approx \boldsymbol{X}_{m\times n} \tag{16}$$

After generating the compressed matrix for the reference summary and candidate summary, a vector for each sentence can be constructed by taking values in the matrix for each term found in that sentence. The cosine distance between the reference vector and the candidate vector can then be calculated as an indication of their semantic similarity.

In our experiment, LSA is trained on the 56 students' summaries. We use a filtering "stop word removal" to exclude common words from the analysis. LSA uses the binary value method to construct a matrix. After preprocessing, SVD is applied to the matrix to represent a matrix into 110 dimensional space. This dimensional space is used for assessing the semantic similarity between two texts.

Bilingual Language Evaluation Understudy (BLEU) (Papineni, Roukos, Ward, & Zhu, 2002) is an n-gram precision based evaluation metric initially designed for the task of machine translation evaluation. The developers of BLEU also suggested that this metric could be used for summarization evaluation – how much the information in a summary text overlaps with the information in a source text using N-gram co-occurrence statistics. BLEU's precision can be computed as the number of words in a candidate summary that matches words in a reference summary divided by the total number of words in the candidate summary. BLEU's n-gram precision is defined as:

$$P_N = \frac{\sum_{S \in \{\text{Candidate summaries}\}} \sum_{\text{N-gram} \in S} \text{Count}_{\text{match}}(\text{N-gram})}{\sum_{S \in \{\text{Candidate summaries}\}} \sum_{\text{N-gram} \in S} \text{Count}(\text{N-gram})} \tag{17}$$

where N is the maximum length of the N-gram, $\text{Count}_{\text{match}}$ (N-gram) is the total number of N-grams co-occurring in a reference summary and a candidate summary, and Count (N-gram) is the number of N-grams in the candidate summary. The N-gram match ratio (Eq. (17)) is used for a comparison of all the sentences in the reference and candidate summaries:

$$\text{BLEU} = BP \times e^{(\sum_{n=1}^{N} W_n \times \log P_n)} \tag{18}$$

$$\text{BP} = \begin{cases} 1 & \text{if } |c| > |r| \\ e^{(1-\frac{r}{c})} & \text{if } |c| \leqslant |r| \end{cases} \tag{19}$$

where $N \geqslant n, n$ and N range from 1 to 4, $W_n = 1/N$. $|c|$ is the length of the candidate summary and $|r|$ is the length of the reference summary. The Brevity Penalty (BP) is calculated to prevent very short candidate summaries from trying to increase their score.

He et al. (2009) proposed a summary assessment system based on the LSA method and N-gram co-occurrence with the aim of assessing students' written summaries. However, a score is assigned to student summary using Eq. (20).

$$Total_{score} = \frac{LSA_{score} + N - gramco - occurrence_{score}}{2} \tag{20}$$

The Automatic Assessment of Students' free-text answers (Pérez et al., 2004) are based on the modified version of BLEU (Papineni et al., 2002) algorithm, called Evaluating Responses with BLEU (ERB), and LSA (Deerwester, Dumais, Landauer, Furnas, & Harshman, 1990; Foltz, Kintsch, & Landauer, 1998). It was developed for grading students' essays. The system compares a student's essay and the model essay to determine how similar they are. Eq. (20) is used to calculate the score of a student's essay.

$$\begin{cases} ERB_{score} = MPB \times e^{\sum_{n=0}^{N} \frac{\log(MUP(n-gam))}{N}} \\ LSA_{score} = \frac{\sum_{\vec{r_i} \in R} \cos(\vec{a}, \vec{r_i})}{2|R|} + 0.5 \\ COMB_{score} = \alpha\, ERB_{score} + (1-\alpha) LSA_{score} \end{cases} \tag{21}$$

where N is the maximum length of the N-gram. The Modified Unified Precision (MUP) (Papineni et al., 2002) metric that clips the frequency of the n-gram according to the number of times it appears in the candidate and in the references. The Modified Brevity Penalty (MBP) is the percentage of the reference text that is covered by the candidate text (Pérez et al., 2004). Let $\vec{a}$ be the document vector obtained from the student summary and $R = \{\vec{r_1}, \vec{r_2}, \ldots, \vec{r_n}\}$ be the set of the document vectors corresponding to the references; α is equal to the 0.5.

5.3. Performance analysis

We conduct our analysis and assess the SALK based on the single-document summarization datasets provided by Document Understanding Conference. The performance of the SALK is compared with other evaluation techniques. We carried out four various tests corresponding to the similar test, dissimilar test, synonym test and grading test for assessing the performance of the SALK in comparison with other assessment techniques. The objectives of these tests are as follows:

Similar test – to determine the ability of the proposed method to provide a high similarity score when the candidate summary and reference similar are related.
Dissimilar test – to determine the ability of the proposed method to provide a low similarity score when the candidate summary and reference dissimilar are unrelated.
Synonym test – to determine the ability of the proposed method to assess the candidate summary based on its content and the different synonym terms in the summary must not have an impact on the performance.
Grading test – to determine the ability of the proposed method to produce a content based similarity score that is close to the score that has been given by a human expert.

- **Evaluation measure**

In this section, we evaluate the results in terms of accuracy. For each test, the accuracy is computed according to Eqs. (22)–(25). We now detail the accuracy rate (AR) corresponding to the similar test, dissimilar test, synonym test and grading test.

Method H_0 – *TextA–textB*. One method that can be used to determine the similarity measure between text A and text B is

as follows. The first, the text A is decomposed into a number of sentences. The second, the similarity measure is calculated between each sentence in text A and all sentences from the text B. The third, the maximum similarity measure is identified, and then it is assigned to the current sentence from text A. Finally, the similarity measure between two texts A and B is calculated by averaging the maximum similarity measures between sentences.

- **Similarity test**

The *similarity score* between a candidate summary and Ref_{sim} is calculated using method H_0. If the similarity score exceeds the threshold, then this similarity score is considered successful. However, the accuracy is calculated using the ratio between the number of candidate summaries for which the similarity scores exceed the threshold and the total number of candidate summaries. The following equation is used to calculate the accuracy rate:

$$\text{AR}_{\text{sim}} = \frac{\text{Total number of summaries that their calculated } \textbf{similarity scores} \text{ exceed threshold}}{\text{Total numbers of summaries}} \tag{22}$$

- **Dissimilarity test**

The dissimilarity score between a candidate summary and Ref_{diss} is calculated similar to the method H_0. If the dissimilarity score does not exceed the threshold, then this dissimilarity score is considered correct. Unlike the similarity text, in the dissimilarity test, the accuracy is calculated using the ratio between the number of candidate summaries for which the dissimilarity scores did not exceed the threshold and the total number of candidate summaries. The following equation is used to calculate the accuracy rate:

$$\text{AR}_{\text{diss}} = \frac{\text{Total number of summaries that their calculated } \textbf{dissimilarity scores do not} \text{ exceed threshold}}{\text{Total numbers of summaries}} \tag{23}$$

- **Synonym test**

The synonym score between a candidate summary and Ref_{syn} is computed using method H_0. If the synonym score exceeds the threshold, then this synonym score is considered successful. However, the accuracy is calculated using the ratio between the number of candidate summaries whose synonym scores exceeded the threshold and the total number of candidate summaries. The following equation is used to calculate the accuracy rate:

$$\text{AR}_{\text{syn}} = \frac{\text{Total number of summaries that their calculated } \textbf{synonym scores} \text{ exceed threshold}}{\text{Total numbers of summaries}} \tag{24}$$

- **Grading test**

A grading score for each student summary is calculated in a similar manner to the human evaluation. We used the holistic grading method (Foltz et al., 1999; León, Olmos, Escudero, Cañas, & Salmerón, 2006) to provide a score for any student summary. In this method, each student summary is assigned a score as follows. First, the similarity measure between a candidate summary (CS) and each of the other candidate summaries is calculated using method H0. Then the two closely similar candidate summaries to the CS are identified. Finally, a score is assigned to the CS by averaging the scores that the human experts had assigned to the closest two. If the assigned score to the CS exceeded the threshold, then this score is considered correct. However, the accuracy is calculated using the ratio between the number of candidate summaries for which the scores exceeded the threshold and the total number of candidate summaries. We use the following equation to calculate the accuracy rate:

$$\text{AR}_{\text{grad}} = \frac{\text{Total number of summaries that their calculated } \textbf{grading scores} \text{ exceed threshold}}{\text{Total numbers of summaries}} \tag{25}$$

Parameter setting – the proposed method requires two parameters to be determined before use: a threshold (Th) to calculate the accuracy rate, and a factor lambda (λ) for weighting the significance between semantic information and syntactic information. Both parameters in the current experiment were found using 300 documents and corresponding summaries. The documents and summaries used for this experiment were taken from DUC 2002.

We ran our proposed method on the current data set. We used Eqs. (22)–(25) to calculate the Accuracy Rate (AR) for four various tests: similarity test (AR_{sim}), dissimilarity test (AR_{diss}), synonym test (AR_{syn}) and grading test (AR_{grad}). We performed each test for each peer lambda between 0.1 and 1 with a step of 0.1 and threshold between 0.1 and 1 with a step of 0.1, (e.g. = 0.3, Th = 0.5). Table 2 presents our experimental results achieved by these tests using various threshold and the values. We evaluate the results in terms of accuracy obtained by each test. We also measure the average accuracy rate using Eq. (26).

$$\textit{Average Accuracy Rate } (\textit{AAR}) = \frac{\text{AR}_{\text{sim}} + \text{AR}_{\text{diss}} + \text{AR}_{\text{syn}} + \text{AR}_{\text{grad}}}{4} \tag{26}$$

By analyzing the results, we find that the best performance is achieved by a threshold = 0.6 and = 0.9. This threshold and produced the accuracy rate for four various tests as follows: 92.54% (Similar), 91.04% (Dissimilar), 81.60% (synonym) and 85.07% (Grading) in term of accuracy. We also got the best average accuracy rate 87.56% by the threshold = 0.6 and = 0.9. The best values of Table 2 have been marked in boldface.

As a result, using the current data set, we obtain the best accuracy of 87.56% when we use 0.6 as the threshold value and 0.9 as the value. Therefore, we can recommend this threshold and lambda for use on the rest of the data set.

Table 2
Performance of the SALK against various threshold and lambda values. (Due to space limitations of this paper, a sample results are shown).

Lambda	Threshold	AR_{sim}	AR_{diss}	AR_{syn}	AR_{grad}	AAR
$\lambda = 0.1$	0.1	1.0000	0.2223	0.9005	1.0000	0.7807
	0.2	0.9255	0.3051	0.9011	1.0000	0.7829
	0.3	0.8010	0.5645	0.8225	1.0000	0.7970
	0.4	0.5205	0.6602	0.5067	0.9510	0.6596
	0.5	0.2619	0.8860	0.2785	0.9333	0.5899
	0.6	0.1705	1.0000	0.0397	0.8321	0.5106
	0.7	0.0000	1.0000	0.0000	0.6198	0.4050
	0.8	0.0000	1.0000	0.0000	0.2200	0.3050
	0.9	0.0000	1.0000	0.0000	0.0330	0.2583
	1	0.0000	1.0000	0.0000	0.0000	0.2500
$\lambda = (0.2, \ldots, 0.8)$	*0.1*	.	.	.	.	.
	.	.	.	.	.	.
	.	.	.	.	.	.
	1	.	.	.	.	.
$\lambda = 0.9$	0.1	1.0000	0.1438	0.9552	1.0000	0.7748
	0.2	1.0000	0.2301	0.9403	1.0000	0.7926
	0.3	0.9850	0.4971	0.9111	1.0000	0.8483
	0.4	0.9701	0.6065	0.8656	1.0000	0.8606
	0.5	0.9300	0.8160	0.8308	0.8801	0.8642
	0.6	**0.9254**	**0.9104**	**0.8160**	**0.8507**	**0.8756**
	0.7	0.8505	0.9752	0.6567	0.7761	0.8146
	0.8	0.6119	1.0000	0.2985	0.4478	0.5896
	0.9	0.2835	1.0000	0.0597	0.0597	0.3507
	1	0.0000	1.0000	0.0000	0.0000	0.2500

Table 3
Performance comparison between SALK and other methods.

Various tests				
Method	Similar AR_{sim}	Dissimilar AR_{diss}	Synonym AR_{syn}	Grading AR_{grad}
SALK	0.8876	0.8576	0.7528	0.8164
LSA	0.7790	0.7715	0.3445	0.6516
BLEU, N (1, . . . , 4)	0.7003	0.7490	0.4382	0.6292
N-gram, N (1, . . . , 4)	0.6479	0.6367	0.5730	0.5393
LSA-ERB	0.8089	0.7977	0.6367	0.6966
LSA-Ngram	0.7135	0.7041	0.4588	0.5955
ROUGE	0.5617	0.7640	0.2771	0.4794

To confirm the aforementioned results, we validate our proposed method, SALK, using a comparison of the overall accuracy obtained by SALK and other existing methods, such as LSA, N-gram,BLEU, LSA_Ngram, LSA_ERB and ROUGE (N-gram, LCS, Skip bigram). We apply these methods to the 267 previously unused documents and corresponding summaries using four different tests only with the threshold value 0.6 and lambda value 0.9. Table 3 and Fig. 6 present the obtained results of accuracy for the four tests with the threshold of 0.6 and the lambda of 0.9. The practical tests prove that SALK outperforms the other examined methods and that it is also more accurate than the other methods. SALK is also able to obtain an accuracy of (82.86%) in comparison with the best existing method, LSA_ERB, which has an accuracy of (73.50%).

- **Detailed comparison**
From the comparison of the accuracy values for other methods, SALK obtains a considerable improvement. Table 4 displays the improvement of SALK for all four various tests. It is clear that SALK achieves the high accuracy and outperforms all the other methods. We use the relative improvement $(\frac{\text{Our method} - \text{Other method}}{\text{Other method}}) \times 100$, for comparison. In Table 4 "+" means the proposed method improves the existing methods. We see that among the existing methods the LSA_ERB displays the best results compared to LSA, N-gram,BLEU, LSA_Ngram and ROUGE(Ngram, LCS, Skipbigram). In comparison with the method LSA_ERB, SALK improves the performance of the LSA_ERB method as follows: 9.73% (Similar test), 7.51% (Dissimilar test), 18.23% (synonym test) and 17.20% (Grading test) in terms of accuracy.

The statistical test (*T*-test) – we use a paired-samples-*T*-test in order to compare the performance of our method with the other methods. The *T*-test is performed on the overall performance measures obtained for each test (i.e., *similarity, dissimilarity, synonym and grading tests*) using an assessment method (e.g. SALK vs BLEU). The study is conducted based on two hypotheses, a) null hypothesis (denoted H_0), our method is not able to obtain high accuracy and improve the

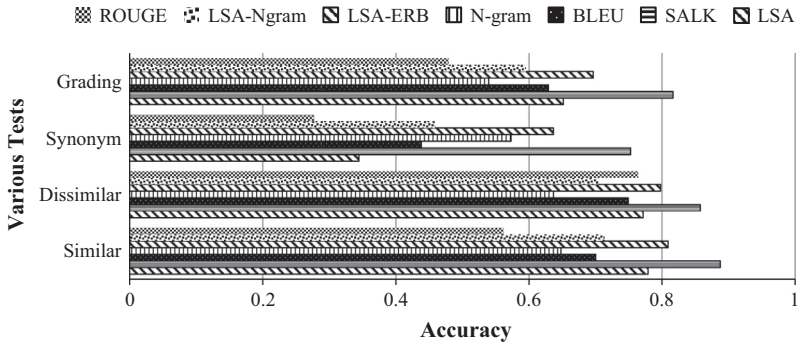

Fig. 6. Performance comparison of the SALK with other methods.

Table 4
Performance evaluation compared between the SALK and other methods.

SALK improvement (%)						
Test	LSA	BLEU	N-gram	LSA-ERB	LSA-Ngram	ROUG
Similar	+13.94	+26.75	+37.00	+9.73	+24.41	+58.02
Dissimilar	+11.16	+14.50	+34.69	+7.51	+22.80	+12.25
Synonym	+118.52	+71.79	+31.38	+18.23	+64.10	+171.67
Grading	+25.29	+29.75	+51.38	+17.20	+37.11	+70.30

Table 5
Statistical comparison between the SALK and other methods.

Our proposed method (SALK) $\alpha = 0.05$								
Method	Similar		Dissimilar		Synonym		Grading	
	T	Sig	*T*	Sig	*T*	Sig	*T*	Sig
LSA	2.805	0.005	4.566	0.000	10.795	0.000	4.940	0.000
BLEU	4.790	0.000	3.637	0.000	8.751	0.000	4.877	0.000
N-gram	5.572	0.000	8.485	0.000	4.965	0.000	7.698	0.000
LSA-ERB	3.153	0.002	2.970	0.003	2.800	0.005	4.374	0.000
LSA-Ngram	4.947	0.000	5.870	0.000	8.935	0.000	5.583	0.000
ROUG	7.091	0.000	3.706	0.000	13.055	0.000	8.045	0.000

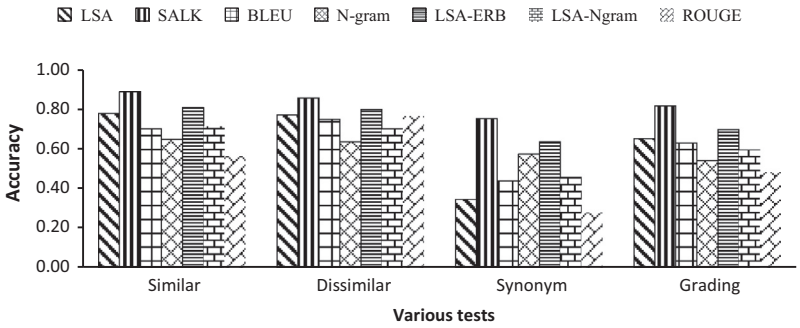

Fig. 7. Comparison accuracy rate between the SALK and other methods.

performance compared with the current methods, and b) the alternative hypothesis (denoted H_A), the proposed method is able to obtain high accuracy and improve the performance compared with the current methods.

Our degrees of freedom (df) are equal to 266. A $t - table$ value with 266 df, for $alpha = 0.05$ is equal to 1.645. The results of the data analyses in Table 5 show that the p-value is very low. However, based on the results (p-value < 0.05), we reject the null hypothesis and accept the alternative hypothesis. In other words, the overall findings of the analysis show that SALK outperforms the other methods.

In order to display the comparison of methods more descriptively, we show it in histograms. The comparison between the overall performance achieved by SALK and other methods for the similar dataset is presented in Fig. 7. This shows that our method obtained better accuracy.

Discussion – from Tables 3 and 4 we make the following main observations. Our method outperforms all other methods. This is due to the fact that, (a) It is able to identify the synonym or similar words among all sentences using a lexical database, Word Net. It is very important to consider this aspect (identifying the synonym or similar words) when evaluating the summaries (Deerwester et al., 1990; Pérez et al., 2005).

(b) Given two sentences (i.e., S_1: *John helps Ravi;* S_2: *Ravi helps John*), unlike ROUGE − S (Alguliev et al., 2011; Ermakova, 2012; Lin, 2004), SALK is able to give credit to S_1, if S_1 does not have any word pair co-occurring with sentence S_2. ROUGE − L (Ermakova, 2012; Lin, 2004) is based on the counting of the main longest common subsequence; therefore, it does not count other alternative and shorter substrings for similarity measuring (Alguliev et al., 2011; Lin, 2004).

(c) N-gram (Lin, 2004) and BLEU (Pérez et al., 2004) calculate the text similarity according to the co-occurring words in the text. They work well in long text unlike the short text; the long text has enough information (i.e., long text has an enough number of co-occurring words). The proposed method improves this limitation.

(d) LSA can produce a reasonable result when it applies to a large corpus (Foltz, 1996; Pérez et al., 2005). LSA uses a predefined word list including hundreds of thousands of words (Landauer, Foltz, & Laham, 1998), for comparison of two texts; this drawback can lead to some important words from the input texts not being considered in the LSA space. While, the proposed method compares two texts on the sentence level based on the words in compared sentences. LSA with high dimensionality and high sparsity has an impact on the performance in similarity measuring (Burgess, Livesay, & Lund, 1998; Salton, 1989). LSA is a 'bag-of-words' method and does not take into account the word order or syntactic information for computing text similarity (Kanejiya et al., 2003; Pérez et al., 2005; Wiemer-Hastings & Zipitria, 2001).

6. Conclusion and future work

We believe that automatic summarization assessment has become an important part of the text summarization. Since summary writing assessment is an arduous and tedious task, we proposed a method (called SALK), which merges semantic and syntactic information in order to produce an effective evaluation method that is able to produce high accuracy. An empirical study was conducted to test SALK. The experimental results of the assessment were satisfactory and provided strong evidence that SALK outperformed other methods. As we already saw, SALK was able to obtain an accuracy of 82.86% in comparison with the best existing technique, (LSA-ERB), which had an accuracy of 73.50%. Moreover, we implemented SALK into an automatic summarization assessment system to grade student written summaries in the English language.

It is worth noting that SALK does not need a deeper linguistic processing than just tokenization, part-of-speech tagging and the lexical database. This helps in keeping the portability across languages that the shallow NLP techniques allow. Further, the common way to assess the content of the summaries is to compare them with a reference summary, which is a hard and expensive task. Much effort is required to have a corpus of texts and their corresponding summaries. To resolve this problem, in our proposed method, a reference summary is no longer necessary, as it takes the original text and summary text as input to assess the summary.

This paper provides the following ideas for future work. Firstly, as SALK only focused on an evaluation of the content, we aim to develop an automatic summarization assessment system by combining an English language assessor, a style checker and additional natural language techniques. Furthermore, we are confident that the idea of incorporating semantic and syntactic information can be further explored, using and combining more complex techniques and modules for text analysis. This is because once a passive or active sentence has been used in writing, it is important to know what passive and active sentences are before comparisons can be drawn. Finally, our method used WordNet as the main semantic knowledge base for the calculation of semantic similarity between words. The comprehensiveness of WordNet is determined by the proportion of words in the text that are covered by its knowledge base. However, the main criticism of WordNet concerns its limited word coverage to calculate semantic similarity between words. Obviously, this disadvantage has a negative effect on the performance of our proposed method. One solution is that, in addition to WordNet, other knowledge resources, such as Wikipedia and other large corpus should be used.

References

Alguliev, R. M., Aliguliyev, R. M., & Mehdiyev, C. A. (2011). Sentence selection for generic document summarization using an adaptive differential evolution algorithm. *Swarm and Evolutionary Computation, 1*, 213–222.

Aliguliyev, R. M. (2009). A new sentence similarity measure and sentence based extractive technique for automatic text summarization. *Expert Systems with Applications, 36*, 7764–7772.
Alonso, L., Castellón, I., Climent, S., Fuentes, M., Padró, L., & Rodrıguez, H. (2004). Approaches to text summarization: Questions and answers. *Inteligencia Artificial, 8*, 22.
Alyousef, H. S. (2006). Teaching reading comprehension to ESL/EFL learners. *Journal of Language and Learning, 5*, 63–73.
Atkinson-Abutridy, J., Mellish, C., & Aitken, S. (2004). Combining information extraction with genetic algorithms for text mining. *IEEE Intelligent Systems, 19*, 22–30.
Aytar, Y., Shah, M., & Luo, J. (2008). Utilizing semantic word similarity measures for video retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008* (pp. 1–8). IEEE.
Brill, E. (1994). Some advances in transformation-based part of speech tagging. arXiv preprint cmp-lg/9406010.
Brown, A. L., & Day, J. D. (1983). Macrorules for summarizing texts: The development of expertise. *Journal of Verbal Learning and Verbal Behavior, 22*, 1–14.
Burgess, C., Livesay, K., & Lund, K. (1998). Explorations in context space: Words, sentences, discourse. *Discourse Processes, 25*, 211–257.
Chang, K.-E., Sung, Y.-T., & Chen, I.-D. (2002). The effect of concept mapping to enhance text comprehension and summarization. *The Journal of Experimental Education, 71*, 5–23.
Cho, Y. (2012). Teaching summary writing through direct instruction to improve text comprehension for students in ESL/EFL classroom. University of Wisconsin-River Falls.
Choi, F. Y., Wiemer-Hastings, P., & Moore, J. (2001). Latent semantic analysis for text segmentation. In Proceedings of EMNLP: Citeseer.
Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., & Harshman, R. A. (1990). Indexing by latent semantic analysis. *JASIS, 41*, 391–407.
De Marneffe, M.-C., MacCartney, B., & Manning, C. D. (2006). Generating typed dependency parses from phrase structure parses. In Proceedings of LREC (vol. 6, pp. 449-454).
Edmundson, H. P. (1969). New methods in automatic extracting. *Journal of the ACM (JACM), 16*, 264–285.
Erkan, G., & Radev, D. R. (2004). LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research (JAIR), 22*, 457–479.
Ermakova, L. (2012). Automatic summary evaluation. Roug e modifications. In VI Российская летняяшкаола по ифмационному поиску (RuSSIR'2012).
Fan, Y.-C. (2010). The effect of comprehension strategy instruction on EFL Learners' reading comprehension. *Asian Social Science, 6*, P19.
Fattah, M. A., & Ren, F. (2009). GA, MR, FFNN, PNN and GMM based models for automatic text summarization. *Computer Speech & Language, 23*, 126–144.
Foltz, P. W. (1996). Latent semantic analysis for text-based research. *Behavior Research Methods, Instruments, & Computers, 28*, 197–202.
Foltz, P. W., Kintsch, W., & Landauer, T. K. (1998). The measurement of textual coherence with latent semantic analysis. *Discourse Processes, 25*, 285–307.
Foltz, P. W., Laham, D., & Landauer, T. K. (1999). The intelligent essay assessor: Applications to educational technology. *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning, 1*.
Franzke, M., & Streeter, L.A. (2006). Building student summarization, writing and reading comprehension skills with guided practice and automated feedback. Highlights From Research at the University of Colorado, A white paper from Pearson Knowledge Technologies.
Hedge, T. (2001). *Teaching and learning in the language classroom* (Vol. 106). Oxford, UK: Oxford University Press.
He, Y., Hui, S. C., & Quan, T. T. (2009). Automatic summary assessment for intelligent tutoring systems. *Computers & Education, 53*, 890–899.
Jones, K. S., & Galliers, J. R. (1996). *Evaluating natural language processing systems: An analysis and review* (Vol. 1083). Springer.
Kanejiya, D., Kumar, A., & Prasad, S. (2003). Automatic evaluation of students' answers using syntactically enhanced LSA. *Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing* (Vol. 2, pp. 53–60). Association for Computational Linguistics.
Ko, Y., Park, J., & Seo, J. (2004). Improving text categorization using the importance of sentences. *Information Processing & Management, 40*, 65–79.
Kupiec, J., Pedersen, J., & Chen, F. (1995). A trainable document summarizer. In *Proceedings of the 18th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 68–73). ACM.
Landauer, T. K. (2002). On the computational basis of learning and cognition: Arguments from LSA. *Psychology of Learning and Motivation, 41*, 43–84.
Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review, 104*, 211.
Landauer, T. K., Laham, D., Rehder, B., & Schreiner, M. E. (1997). How well can passage meaning be derived without using word order? A comparison of Latent Semantic Analysis and humans. In *Proceedings of the 19th annual meeting of the cognitive science society* (pp. 412–417).
Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes, 25*, 259–284.
Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 159–174.
León, J. A., Olmos, R., Escudero, I., Cañas, J. J., & Salmerón, L. (2006). Assessing short summaries with human judgments procedure and latent semantic analysis in narrative and expository texts. *Behavior Research Methods, 38*, 616–627.
Li, Y., McLean, D., Bandar, Z. A., O'shea, J. D., & Crockett, K. (2006). Sentence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge and Data Engineering, 18*, 1138–1150.
Lin, D. (1998). An information-theoretic definition of similarity. In *ICML* (Vol. 98, pp. 296–304).
Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop* (pp. 74–81).
Mihalcea, R., Corley, C., & Strapparava, C. (2006). Corpus-based and knowledge-based measures of text semantic similarity. In *AAAI* (Vol. 6, pp. 775–780).
Miller, G. A., & Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes, 6*, 1–28.
Mohler, M., Bunescu, R., & Mihalcea, R. (2011). Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies* (Vol. 1, pp. 752–762). Association for Computational Linguistics.
Pakzadian, M., & Rasekh, A. E. (2013). The effects of using summarization strategies on Iranian EFL learners' reading comprehension. *English Linguistics Research, 1*, p118.
Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics* (pp. 311–318). Association for Computational Linguistics.
Pérez, D., Alfonseca, E., & Rodrıguez, P. (2004). Upper bounds of the BLEU algorithm applied to assessing student essays. In *Proceedings of the 30th international association for educational assessment (IAEA) conference*.
Pérez, D., Gliozzo, A.M., Strapparava, C., Alfonseca, E., Rodríguez, P., & Magnini, B. (2005). Automatic Assessment of students' free-text answers underpinned by the combination of a BLEU-inspired algorithm and latent semantic analysis. In *FLAIRS conference* (pp. 358–363).
Rus, V., Graesser, A., & Desai, K. (2007). Lexico-syntactic subsumption for textual entailment. *Amsterdam Studies in the Theory and History of Linguistic Science Series, 4*, 292. 187.
Salton, G. (1989). Automatic text processing: The transformation, analysis, and retrieval of: Addison-Wesley.
Tian, Y., Li, H., Cai, Q., & Zhao, S. (2010). Measuring the similarity of short texts by word similarity and tree kernels. In *Information Computing and Telecommunications (YC-ICT), 2010 IEEE Youth Conference on* (pp. 363–366). IEEE.
Valenti, S., Neri, F., & Cucchiarelli, A. (2003). An overview of current research on automated essay grading. *Journal of Information Technology Education: Research, 2*, 319–330.
Warin, M. (2004). Using WordNet and semantic similarity to disambiguate an ontology. (Retrieved 25.01.08).
Wiemer-Hastings, P., & Wiemer, P. (2000). Adding syntactic information to LSA. In *Proceedings of the 22nd annual meeting of the cognitive science society*. Citeseer.
Wiemer-Hastings, P., & Zipitria, I. (2001). Rules for syntax, vectors for semantics. In *Proceedings of the twenty-third annual conference of the cognitive science society* (pp. 1112–1117).

Zhang, J., Sun, L., & Zhou, Q. (2005). A cue-based hub-authority approach for multi-document text summarization. In *Proceedings of 2005 IEEE international conference on natural language processing and knowledge engineering, 2005. IEEE NLP-KE'05* (pp. 642–645). IEEE.

Zipitria, I., Elorriaga, J. A., Arruarte, A., & de Ilarraza, A. D. (2004). From human to automatic summary evaluation. In *Intelligent tutoring systems* (pp. 432–442). Springer.

Zipitria, I., Arruarte, A., Elorriaga, J. A. (2010). Automatically grading the use of language in learner summaries. In *Proceedings of the 18th international conference on computers in education, Putrajaya, Malaysia* (pp. 46–50).

Zipitria, I., Larrañaga, P., Armañanzas, R., Arruarte, A., & Elorriaga, J. A. (2008). What is behind a summary-evaluation decision? *Behavior Research Methods, 40*, 597–612.